# Pentium® Pro Processor Specification Update

Release Date: February, 1996

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained from:

Intel Corporation
P.O. Box 7641
Mt. Prospect, IL 60056-7641

or call 1-800-879-4683

# CONTENTS

## REVISION HISTORY

| Date of Revision | Version | Description |
|---|---|---|
| November, 1995 | -001 | This document is the first Specification Update for the Intel Pentium® Pro processor. |
| December, 1995 | -002 | Added Errata 29-38, and 3AP. Included additional S-specs. Added a case to Erratum 3. Clarified Errata 2, 6, 8, 10, 11, and 26. |
| January, 1995 | -003 | Added Errata 39-42. Updated the status for Erratum 10. Added documentation for 166-, 180-, and 200-MHz processors. |
| February, 1995 | -004 | Clarified Errata 11, 22, 30, and 34. Updated status for Errata 10, 15, 40, and 42. Added CPUID information for determining L2 cache size. Added S-specs and errata column for sA1 stepping. |

# PREFACE

This document is an update to the specifications contained in the *Pentium® Pro Processor Developer's Manual*, *Volumes 1, 2, and 3* (Order Numbers 242690, 242691, and 242692, respectively). It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains Specification Changes, S-Specs, Errata, Specification Clarifications, and Documentation Changes.

## *Nomenclature*

**S-Specs** are exceptions to the published specifications, and apply only to the units assembled under that s-spec.

**Specification Changes** are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

**Specification Clarifications** describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

**Documentation Changes** include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

**Errata** are design defects or errors. These may cause the microprocessor's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

## *Identification Information*

The Pentium® Pro processor can be identified by the following values:

| Family[1] | 150-, 166-, 180-, and 200-MHz Model 1[2] |
|---|---|
| 0110 | 0001 |

**NOTES:**

1 The Family corresponds to bits [11:8] of the EDX register after RESET, bits [11:8] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the generation field of the Device ID register accessible through Boundary Scan.

2 The Model corresponds to bits [7:4] of the EDX register after RESET, bits [7:4] of the EAX register after the CPUID instruction is executed with a 1 in the EAX register, and the model field of the Device ID register accessible through Boundary Scan.

The Pentium Pro processor's level 2 (L2) cache size can be determined by the following values:

| 256-Kbyte Unified L2 Cache[1] | 512-Kbyte Unified L2 Cache[1] |
|---|---|
| 42h | 43h |

**NOTES:**

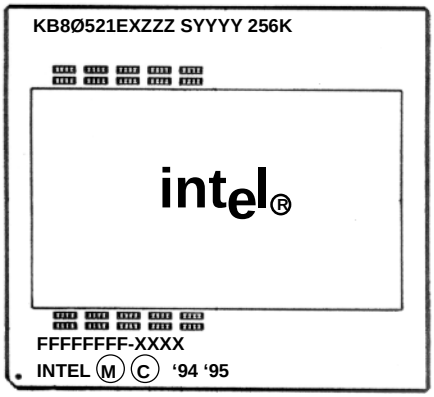
1 The unified L2 cache size corresponds to bits [3:0] of the EDX register after the CPUID instruction is executed with a 2 in the EAX register.

# Specification Update for 150-, 166-, 180-, and 200-MHz Pentium® Pro Processors

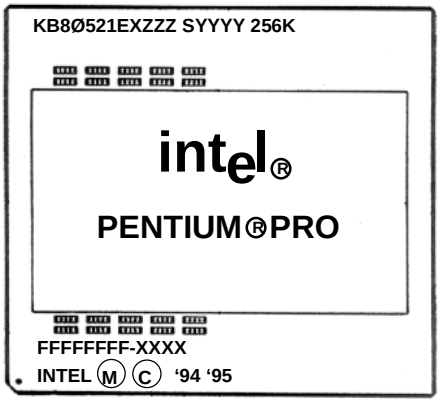
# GENERAL INFORMATION

## *Top Markings*

**B-Step Production Units – Top:**

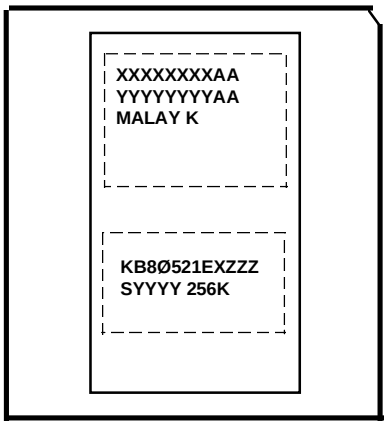



**B-, C-, and sA-Step Production Units – Top:**




## *Bottom Markings*

**B-, C-, and sA-Step Production Units – Bottom:**




**NOTES:**

- ZZZ = Speed (MHz).
- QYYYY = Sample Specification Number.
- SYYYY = S-spec Number.
- FFFFFFFF = FPO # (Test Lot Traceability #).
- XXXX = Serialization Code.
- XXXXXXXXAA = Alternative Identification Number.
- YYYYYYYYAA = Alternative identification Number.
- Top side: inner line indicates boundary of heat spreader.

**Basic 150-, 166-, 180-, and 200-MHz Pentium® Pro Processor Identification Information**

| CPUID | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Type | Family | Model | Stepping | Mfg. Stepping | Speed (MHz) Core/Bus | S-Spec | $V_{CC}$ | $T_{CASE}$ | Notes |
| 0 | 6 | 1 | 1 | B0 | 150/60 | SY002 | 3.1V ± 5% | 0°C - 85°C | 1 |
| 0 | 6 | 1 | 1 | B0 | 150/60 | SY011 | 3.1V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 1 | B0 | 150/60 | SY014 | 3.1 V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 2 | C0 | 150/60 | SY010 | 3.1V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 6 | sA0[2] | 180/60 | SY012 | 3.3V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 6 | sA0[2] | 200/66 | SY013 | 3.3V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 6 | sA0[2] | 166/66 | SY024 | 3.3V ± 5% | 0°C - 85°C | 3 |
| 0 | 6 | 1 | 6 | sA0[2] | 200/66 | SY025 | 3.3V ± 5% | 0°C - 85°C | 3 |
| 0 | 6 | 1 | 7 | sA1 | 200/66 | SY031 | 3.3V ± 5% | 0°C - 85°C | |
| 0 | 6 | 1 | 7 | sA1 | 180/60 | SY032 | 3.3V ± 5% | 0°C - 85°C | |

**NOTES:**

1. The VID pins are not supported on these parts.
2. The sA0 stepping is logically equivalent to the C0 stepping, but on a different manufacturing process.
3. These parts are equipped with a 512-Kbyte L2 cache.

## *Summary Table of Changes*

The following table indicates the Specification Changes, Errata, Specification Clarifications, or Documentation Changes which apply to the 150-, 166-, 180-, and 200-MHz Pentium® Pro processors. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

**CODES USED IN SUMMARY TABLE**

| | |
|---|---|
| X: | Erratum which applies to the given stepping. |
| Doc: | Document change or update that will be implemented. |
| Fix: | This erratum is intended to be fixed in a future stepping of the component. |
| Fixed: | This erratum has been previously fixed. |
| NoFix: | There are no plans to fix this erratum. |
| (No mark) or (blank box): | This erratum is fixed in the given stepping or this specification change does not apply to the given stepping. |
| AP: | APIC related erratum. |
| Shaded: | This erratum is either new or modified from the previous version of the document. |

| NO. | B0 | C0 | sA0 | sA1 | PLANS | ERRATA |
|---|---|---|---|---|---|---|
| 1 | X | X | X | X | NoFix | Mixed cacheability of lock variables is problematic in MP systems |
| 2 | X | X | X | X | NoFix | FEA incorrectly calculated after FP access which wraps 64-Kbyte boundary in 16-bit code |
| 3 | X | X | X | X | NoFix | Differences exist in debug exception reporting |
| 4 | X | X | X | X | NoFix | FLUSH# servicing delayed while waiting for STARTUP_IPI in MP systems |
| 5 | X | | | | Fixed | Fast Strings REP MOVS may not transfer all data |
| 6 | X | | | | Fixed | Page table base change during task switch using Mode C paging may corrupt EIP |
| 7 | X | X | X | X | NoFix | Code fetch matching disabled debug register may cause debug exception |
| 8 | X | X | X | X | NoFix | Mode C paging in SMM causes use of incorrect page tables |
| 9 | X | | | | Fixed | Memory indirect near call may corrupt EIP |
| 10 | X | | | | Fixed | L2 single-bit correctable error may cancel simultaneous valid data |
| 11 | X | | | | Fixed | Page split access before write to CR3 may cause hang |
| 12 | X | | | | Fixed | Active A20M# during SMM dump |
| 13 | X | | | | Fixed | Split access across 4-Kbyte page boundary may cause hang |
| 14 | X | X | X | X | NoFix | Checker BIST failure in FRC mode not signaled |
| 15 | X | X | X | X | NoFix | BINIT# assertion causes FRCERR assertion in FRC mode |
| 16 | X | X | X | X | Fix | Extra page fault may occur on IRET during task switch |
| 17 | X | | | | Fixed | Some caching models in SMM may cause shutdown |
| 18 | X | | | | Fixed | Fast Strings feature re-enabled after INIT event |

| NO. | B0 | C0 | sA0 | sA1 | PLANS | ERRATA |
|---|---|---|---|---|---|---|
| 19 | X | | | | Fixed | THERMTRIP# feature not present |
| 20 | X | X | X | X | Fix | OUT instruction, Branch Trace Messages may write incorrect data |
| 21 | | X | X | X | Fix | THERMTRIP# pin not asserted for catastrophic thermal condition |
| 22 | X | X | X | X | Fix | LBER value may not be updated correctly |
| 23 | X | X | X | X | NoFix | BTM for SMI will contain incorrect FROM EIP |
| 24 | X | X | X | X | NoFix | Task switch fault may allow read access of linear address 0h |
| 25 | X | | | | Fixed | Low frequencies with 5:2 core to bus clock ratio may fail in FRC |
| 26 | X | X | X | X | Fix | RDPMC cannot be used in conjunction with SMM |
| 27 | X | X | X | X | Fix | PWRGOOD forced to 0 during boundary scan resets TAP |
| 28 | X | X | X | X | Fix | BIST failure not indicated when RUNBIST TAP command used |
| 29 | X | X | X | X | Fix | INVLPG may not invalidate targeted ITLB entry |
| 30 | X | X | X | X | Fix | SMI does not flush TLB entries with PGE enabled |
| 31 | X | X | X | X | NoFix | I/O restart in SMM may fail after simultaneous MCE |
| 32 | X | X | X | X | Fix | SMBASE reset on INIT# or INIT_IPI |
| 33 | X | X | X | X | Fix | MCE due to L2 ECC error gives L1 MCACOD.LL |
| 34 | X | X | X | X | Fix | INVLPG does not invalidate entire 0 to 4-Mbyte region |
| 35 | X | | | | Fixed | BINIT# assertion during snoop hit may cause double Machine Check Exception |
| 36 | X | X | X | X | NoFix | Machine Check Exception handler may not always execute successfully |
| 37 | X | X | X | X | Fix | Double ECC error on read may result in BINIT# |
| 38 | X | X | X | X | Fix | RSM cannot return to HALT or SHUTDOWN in 32-Bit OS |
| 39 | X | X | X | | Fixed | Accesses of Modified data may hang system |
| 40 | X | X | X | X | NoFix | Branch traps do not function if BTMs are also enabled |
| 41 | X | X | X | | Fixed | Cache line may exist in two different ways in MP system |
| 42 | X | X | X | X | Fixed | HALT, SHUTDOWN, and STPCLK special cycles not issued |
| 1AP | X | X | X | X | NoFix | APIC access to cacheable memory causes shutdown |
| 2AP | X | X | X | X | NoFix | MP systems may hang due to catastrophic errors during BSP determination |
| 3AP | X | X | X | X | Fix | INIT_IPI after STARTUP_IPI-STARTUP_IPI sequence may cause AP to execute at 0h |

# ERRATA

## 1. Mixed Cacheability of Lock Variables Is Problematic in MP Systems

**PROBLEM:** This errata only affects multiprocessor systems where a lock variable address is marked cacheable in one processor and uncacheable in any others. The processors which have it marked uncacheable may stall indefinitely when accessing the lock variable. The stall is only encountered if:

- One processor has the lock variable cached, and is attempting to execute a cache lock.
- If the processor which has that address cached has it cached in the L2 only.
- Several other processors, meanwhile, issue back to back accesses to that same address on the bus.

**IMPLICATION:** MP systems where all processors either use cache locks or consistent locks to uncacheable space will not encounter this problem. If, however, a lock variable's cacheability varies in different processors, and several processors are all attempting to perform the lock simultaneously, an indefinite stall may be experienced by the processors which have it marked uncacheable in locking the variable (if the conditions above are satisfied). Intel has only encountered this problem in focus testing with artificially generated external events. Intel has not currently identified any commercial software which exhibits this problem.

**WORKAROUND:** Follow a homogenous model for the memory type range registers (MTRRs), ensuring that all processors have the same cacheability attributes for each region of memory; do not use locks whose memory type is cacheable on one processor, and uncacheable on others. Avoid page table aliasing, which may produce a non-homogenous memory model.

## 2. FEA Incorrectly Calculated After FP Access Which Wraps 64-Kbyte Boundary in 16-Bit Code

**PROBLEM:** The FEA (floating-point effective address) is the effective address associated with the last non-transparent floating-point instruction executed by the machine. If an 80-bit floating-point access (load or store) occurs in 16-bit mode which accesses memory across a 64-Kbyte boundary, and a subsequent floating-point store occurs in 32-bit mode, the subtraction routine used to calculate the FEA will assume the previous (80-bit) floating-point access was in 32-bit mode, and the high word of the address will be FFFFh instead of 0000h.

**IMPLICATION:** A 32-bit operating system running 16-bit code may encounter this erratum, under the following conditions:

- The operating system is using a segment greater than 64-Kbytes in size.
- An 80-bit floating-point load which wraps the 64-Kbyte boundary is successfully executed.
- The operating system uses a 32-bit handler on an unmasked exception which occurs during the load.

Wrapping an 80-bit floating-point load around the segment boundary in this way is not a normal or useful programming practice. Intel has not currently identified any software which exhibits this behavior.

**WORKAROUND:** If FEA is to be used in 32-bit code run in conjunction with 16-bit floating-point code, care must be taken to ensure that no 80-bit floating-point accesses are split across a 64-Kbyte boundary.

## 3. Differences Exist in Debug Exception Reporting

**PROBLEM:** There exist some differences in the reporting of code and data breakpoint matches between that specified by previous Intel processors' specifications and the behavior of the Pentium Pro processor, as described below:

**CASE 1:**

The first case is for a breakpoint set on a MOVSS or POPSS instruction, when the instruction following it causes a debug register protection fault (DR7.gd is already set, enabling the fault). The Pentium processor reports delayed data breakpoint matches from the MOVSS or POPSS instructions by setting the matching DR6.bi bits, along with the debug register protection fault (DR6.bd). If additional breakpoint faults are matched during the call of the debug fault handler, the Pentium processor sets the breakpoint match bits (DR6.bi) to reflect the breakpoints matched by both the MOVSS or POPSS breakpoint and the debug fault handler call. The Pentium Pro processor only sets DR6.bd in either situation, and does not set any of the DR6.bi bits.

**CASE 2:**

In the second breakpoint reporting failure case, if a MOVSS or POPSS instruction with a data breakpoint is followed by a store to memory which crosses a 4-Kbyte page boundary, the breakpoint information for the MOVSS or POPSS will be lost. Previous processors retain this information across such a page split.

**CASE 3:**

If they occur after a MOVSS or POPSS instruction, the INT *n*, INTO, and INT3 instructions zero the DR6.Bi bits (bits B0 through B3), clearing pending breakpoint information, unlike previous processors.

**CASE 4:**

If a data breakpoint and an SMI (System Management Interrupt) occur simultaneously, the SMI will be serviced via a call to the SMM handler, and the pending breakpoint will be lost.

**IMPLICATION:** When debugging or when developing debuggers for a Pentium Pro processor system, this behavior should be noted. Normal usage of the MOVSS or POPSS instructions (i.e. following them with a MOV ESP) will not exhibit the behavior of cases 1-3. Debugging in conjunction with SMM will be limited by case 4 (no workaround has been identified for this case).

**WORKAROUND:** Following MOVSS and POPSS instructions with a MOV ESP instruction when using breakpoints will avoid the first three cases of this erratum.

## 4. FLUSH# Servicing Delayed While Waiting for STARTUP_IPI in MP Systems

**PROBLEM:** In a multiprocessor system, if an application processor is waiting for a startup interprocessor interrupt (STARTUP_IPI), then it will not service a FLUSH# pin assertion until it has received the STARTUP_IPI.

**IMPLICATION:** After the MP initialization protocol, only one processor becomes the bootstrap processor (BSP). All others become slave application processors (APs). After losing the BSP arbitration, APs go into a wait loop waiting for a STARTUP_IPI.

The BSP can wake up an AP to perform some tasks with a STARTUP_IPI, and then put one or more APs back to sleep with an initialization interprocessor interrupt (INIT_IPI, which has the same affect as asserting INIT#), which returns them to a wait loop. The result is a possible loss of cache coherency if the offline

processor is intended to service a FLUSH# assertion at this point. The FLUSH# will be serviced as soon as the processor is awakened by a STARTUP_IPI, before any other instructions are executed. Intel has not encountered any operating systems that are affected by this problem.

**WORKAROUND:** Operating system developers should take care to execute a WBINVD instruction before an AP is taken offline using an INIT_IPI.

## 5. Fast Strings REP MOVS May Not Transfer All Data

**PROBLEM:** If the fast strings feature is enabled (bit 2 of MSR 1E0h is 0), a REP MOVS instruction may incorrectly use a pre-decremented counter, thereby skipping a cache line of the transfer. This will only occur if paging is enabled.

**IMPLICATION:** A REP MOVS instruction executed with this feature enabled may not transfer all requested data, leading to stale data being left at the destination.

**WORKAROUND:** For steppings affected by this erratum, the fast strings feature must be disabled in the BIOS, by setting bit 2 of the machine specific register (MSR) at address 1E0h to '1'.

## 6. Page Table Base Change During Task Switch Using Mode C Paging May Corrupt EIP

**PROBLEM:** While using mode C paging (a 36-bit address extension which uses 2-Mbyte pages), if data is written to CR3 during a task switch, the EIP will be corrupted, resulting in unpredictable failure. CR3, which contains the page table base, will be written during a task switch if the switch is moving to a new paging environment, or if the TLB entries must be cleared.

**IMPLICATION:** An operating system which uses a hardware tasking model (via the task switch segment, or TSS) along with the 36-bit extension mode C paging, may fail, most likely with an invalid opcode exception. Intel has not identified any operating systems which use this combination of features.

**WORKAROUND:** When developing an operating system which uses mode C paging, do not use hardware task switching, or else only perform hardware task switches which do not change the page table base register, CR3.

## 7. Code Fetch Matching Disabled Debug Register May Cause Debug Exception

**PROBLEM:** If any debug breakpoints are enabled for an instruction, a debug exception is signaled. If the code fetch for a subsequent instruction matches one of the debug registers (DR0 through DR3), and the instruction encounters a page fault, then a debug exception is signaled prior to the page fault even if the code breakpoint is disabled.

**IMPLICATION:** While debugging software, extraneous debug exceptions may be encountered if these registers are not cleared. The debug exception will be serviced, and if the resume flag (RF) is not set in the EFLAGS register, the page fault and debug exception will occur again upon returning to the instruction.

**WORKAROUND:** Set RF in the EFLAGS register at the end of the debug exception handler to avoid multiple breakpoints on one instruction, or ensure that the debug registers are cleared after each debug session.

## 8. Mode C Paging in SMM Causes Use of Incorrect Page Tables

**PROBLEM:** If mode C paging (a 36-bit address extension which uses 2-Mbyte pages) is in use, an SMI is serviced, and the SMM handler switches to mode C paging with a different set of page tables, then upon returning from the SMM handler via an RSM instruction, the processor will use the wrong page tables for address translation. Since paging is not, in general, supported in SMM, this is not a violation of the Pentium Pro processor specification, and has been removed as an erratum, but has been included here for continuity.

**IMPLICATION:** Using mode C paging both in normal operation and in System Management Mode (SMM) may cause the system to hang. Intel has not currently identified any software which is affected by this erratum. This is not a violation of the Pentium Pro processor specification; large size pages are not supported in SMM.

**WORKAROUND:** Do not use mode C paging for the SMM handler code.

## 9. Memory Indirect Near Call May Corrupt EIP

**PROBLEM:** Under some conditions, a near indirect call (opcode FF /2) may cause corruption of the EIP after partial completion of the instruction's execution. This can occur if an access due to the near indirect call:

- Accesses memory which crosses a page boundary.
- Results in a misaligned data breakpoint exception.
- Is the first access of the page containing the data used for the call.
- Results in a page fault which, upon walking the page tables (see Erratum #13), is found not to be a faulting condition.

**IMPLICATION:** The most likely failure mechanism for this erratum is an invalid opcode exception.

**WORKAROUND:** It is possible for BIOS code to contain a workaround for this erratum.

## 10. L2 Single-Bit Correctable Error May Cancel Simultaneous Valid Data

**PROBLEM:** If a single-bit correctable error is detected by the Pentium Pro processor on a transaction between the data cache unit (DCU) of the CPU and the L2 cache, and there is a second transaction pending which caches data from the Pentium Pro processor bus, the Pentium Pro processor will cancel both transactions instead of just the transaction on which the error occurred.

**IMPLICATION:** The canceled transaction will never complete under these conditions. Single-bit errors which encounter this boundary condition will cause the Pentium Pro processor to hang. However, since this erratum only affects error recovery, normal, hardware error-free execution will not be affected.

**WORKAROUND:** None identified.

## 11. Page Split Access Before Write to CR3 May Cause Hang

**PROBLEM:** If paging is enabled, and a memory access occurs which crosses a page boundary, a future write to CR3 may cause the processor to hang. The write to CR3 must occur at a specific time interval away from the page split access; in focus-testing, the only test which encountered this hang in mode A or B paging had 12 NOP instructions between the split access and the write to CR3. When using mode C paging (2-Mbyte pages), the probability of encountering this erratum increased somewhat.

**IMPLICATION:** Using paging in conjunction with writes to CR3 (to invalidate the page tables or change to a new paging environment) may cause the system to hang. There is a small, timing dependent window of opportunity to encounter this erratum, which is wider if mode C paging is used.

**WORKAROUND:** Some instructions can be inserted between the memory access and the write to CR3 to prevent this erratum, such as:

- CPUID,
- STD,
- CLD, or
- CLI/STI.

Though the following instructions between the memory access and the write to CR3 prevent the erratum, they may not be suitable for all situations as workarounds for this erratum due to their effect or dependency on system state:

- WRMSR,
- RDMSR,
- RDPMC,
- LDS, LFS, LGS, LSS, or LES,
- MOV CR0, reg,
- MOV CR2, reg,
- MOV CR4, reg,
- MOV DS, reg,
- MOV SS, reg,
- MOV ES, reg,
- MOV GS, reg,
- MOV FS, reg,
- A far transfer, interrupt, or software interrupt.

Any of these which occur between the memory access which crosses a 4-Kbyte boundary and the write to CR3 will prevent the conditions necessary for this erratum to occur. Due to its long latency, a non-string OUT instruction will also prevent the erratum.

## *12. Active A20M# During SMM Dump*

**PROBLEM:** If the A20M# signal (mask A20) is active when the SMM handler is saving CPU state information to the SMM dump space prior to executing the handler code, the state information will be partially saved at an address with the A20 pin masked, and partially with A20 unmasked.

**IMPLICATION:** If the SMM dump space uses addresses in an odd megabyte (A20 high), the stored state will be split among two different address spaces and the system will not recover state correctly after an RSM instruction.

**WORKAROUND:** Either ensure that the SMM dump space does not coincide with any odd-megabyte addresses (under 1-Mbyte is acceptable), or use external logic to prevent the servicing of an SMI until A20M# is deasserted.

## 13. Split Access Across 4-Kbyte Page Boundary May Cause Hang

**PROBLEM:** When a load or store access is split across a page boundary, a page fault may be detected. If a page fault is signaled during such an access, the Pentium Pro processor will walk the page table to ensure that the page is not being made accessible by another agent (such as another processor in an MP system, or a DMA controller) before branching to the page fault handler. If the split access signals a page fault, and the page which caused the fault is then found to be non-faulting during the page table walk, the access will be executed incorrectly, resulting in a hang.

**IMPLICATION:** When multiple agents are capable of changing the page table, a load or store which is split across a page boundary may cause the machine to hang.

**WORKAROUND:** It is possible for BIOS code to contain a workaround for this erratum.

## 14. Checker BIST Failure in FRC Mode Not Signaled

**PROBLEM:** If a system is running in functional redundancy checking (FRC) mode, and the checker of the master-checker pair encounters a hard failure while running the built-in self test (BIST), the checker will tri-state all outputs without signaling an IERR#.

**IMPLICATION:** Assuming the master passes BIST successfully, it will continue execution unchecked, operating without functional redundancy. However, the necessary pull-up on the FRCERR pin will cause an FRCERR to be signaled. The operation of the master depends on the implementation of FRCERR.

**WORKAROUND:** For successful detection of BIST failure in the checker of an FRC pair, the FRCERR signal must be used, instead of IERR#.

## 15. BINIT# Assertion Causes FRCERR Assertion in FRC Mode

**PROBLEM:** If a pair of Pentium Pro processors are running in functional redundancy checking (FRC) mode, and a catastrophic error condition causes BINIT# to be asserted, the checker in the master-checker pair will enter shutdown. The next bus transaction from the master will then result in the assertion of FRCERR.

**IMPLICATION:** Bus initialization via an assertion of BINIT# occurs as the result of a catastrophic error condition which precludes the continuing reliable execution of the system. Under normal circumstances, the master-checker pair would remain synchronized in the execution of the BINIT# handler. However, due to this erratum, an FRCERR will be signaled. System behavior then depends on the implementation of FRCERR.

**WORKAROUND:** None identified.

## 16. Extra Page Fault May Occur on IRET During Task Switch

**PROBLEM:** During a hardware task switch (using the task switch segment, or TSS) which jumps to the new task with an IRET which causes a page fault, a second page fault exception may be generated upon completion of the page miss handler routine, prompting the Pentium Pro processor to execute the page miss handler a second time.

**IMPLICATION:** Intel has identified no adverse implications of this anomaly other than the slight performance loss from executing the page miss handler twice after a task switch which causes a page fault.

**WORKAROUND:** None identified.

## 17. Some Caching Models in SMM May Cause Shutdown

**PROBLEM:** A condition exists in the Pentium Pro processor such that upon returning from the SMM handler (via an RSM instruction), a read may be executed too soon for the data to be successfully loaded. This condition occurs whenever the load for the read is delayed because the cache line which the load accesses is not present in the cache. This may occur if:

- The SMM handler is too large to fit into the cache.
- A WBINVD instruction was executed upon servicing the SMI.
- The SMM handler is being run in memory mapped with the UC (uncacheable) memory type.

When the RSM instruction is executed under any of these conditions, the first memory access after the RSM will cause multiple stack faults, and the Pentium Pro processor will enter shutdown.

**IMPLICATION:** Under many operating conditions, returning from SMM will result in the CPU entering shutdown.

**WORKAROUND:** It is possible for BIOS code to contain a workaround for this erratum.

## 18. Fast Strings Feature Re-enabled After INIT Event

**PROBLEM:** If the Fast Strings feature has been disabled in the machine specific registers (MSRs) by clearing the appropriate bit (to '0'), after an INIT or an initialization interprocessor interrupt (INIT_IPI), the bit will be set (to '1'), and the feature will be re-enabled.

**IMPLICATION:** After each INIT or INIT_IPI event, this feature will be re-enabled, exposing the system to the effects of Erratum #5.

**WORKAROUND:** It is possible for BIOS code to contain a workaround for this erratum.

## 19. THERMTRIP# Feature Not Present

**PROBLEM:** The THERMTRIP# feature documented for the Pentium Pro processor is not present.

**IMPLICATION:** The Pentium Pro processor will not shut down upon reaching a catastrophic thermal condition as defined in the *Pentium® Pro Processor Developer's Manual, Volume 1.*

**WORKAROUND:** System manufacturers can implement a thermal management solution which does not rely upon THERMTRIP#.

## 20. OUT Instruction, Branch Trace Messages May Write Incorrect Data

**PROBLEM:** A data cache load may be blocked by internal conditions, and speculatively awakened by the Pentium Pro processor. If this speculative wakeup occurs at the same time that an OUT instruction or a branch trace message (BTM) is dispatched, and the load access crosses a cache line boundary, the buffer which contains the data for the OUT or BTM will be corrupted.

**IMPLICATION:** Some OUT instructions and BTMs may propagate corrupted data.

**WORKAROUND:** It is possible for BIOS code to contain a workaround for this erratum.

## 21. THERMTRIP# Pin Not Asserted For Catastrophic Thermal Condition

**PROBLEM:** If a catastrophic thermal condition is reached as specified in the *Pentium® Pro Processor Developer's Manual, Volume 1,* the processor will correctly shut down until a hard reset is performed. However, the THERMTRIP# pin, which should be asserted after the processor shuts down, will remain unasserted.

**IMPLICATION:** If the processor shuts down due to a catastrophic thermal condition, there will be no electrical indication of the reason for the shutdown, since the THERMTRIP# pin will not be asserted.

**WORKAROUND:** An external temperature sensor should be implemented if a system design requires notification that a catastrophic thermal condition has caused the system to shut down.

## 22. LBER Value May Not Be Updated Correctly

**PROBLEM:** The last branch record (LBR) and the last branch before exception record (LBER) can be used to determine the source and destination information for previous branches or exceptions. The LBR contains the source and destination addresses for the last branch or exception, and the LBER contains similar information for the last branch taken before the last exception. This information is typically used to determine the location of a branch which leads to execution of code which causes an exception. However, after executing a trap under some conditions, the LBER may not be updated after a branch, or may be updated incorrectly.

**IMPLICATION:** The LBER and LBR registers are used only for debugging purposes. When this erratum occurs, the LBER will not contain reliable address information. The value of LBER should be used with caution when debugging branching code; if the values in the LBR and LBER are the same, the value of LBER is 0, or the value of the LBER is the same before and after the trap, then the LBER value is incorrect.

**WORKAROUND:** None identified.

## 23. BTM for SMI Will Contain Incorrect FROM EIP

**PROBLEM:** A system management interrupt (SMI) will produce a branch trace message (BTM), if BTMs are enabled. However, the FROM EIP field of the BTM (used to determine the address of the instruction which was being executed when the SMI was serviced) will not have been updated for the SMI, so the field will report the same FROM EIP as the previous BTM.

**IMPLICATION:** A BTM which is issued for an SMI will not contain the correct FROM EIP, limiting the usefulness of BTMs for debugging software in conjunction with System Management Mode (SMM).

**WORKAROUND:** None identified.

## 24. Task Switch Fault May Allow Read Access of Linear Address 0h

**PROBLEM:** For this erratum to occur, the following conditions must be satisfied:

- The task switch segment (TSS) must be used to handle task switches.
- A task switch fault must occur while loading the new segment descriptors for a task switch.
- The resulting task switch fault handler call must not result in a task switch.
- The task switch fault handler must be executed successfully and return to the task switch using a far return or an IRET instruction.

Upon the coincidence of these conditions, any of the data segment registers which were not loaded by the task switch will permit a read access to a single byte at linear address 0h.

**IMPLICATION:** Under these circumstances, if the target task is of a privilege level of 1, 2, or 3, it may be allowed to read the byte at linear address 0h, which it may not normally be permitted to read. Intel has not currently identified any software which could cause this behavior.

**WORKAROUND:** When developing an operating system which uses the TSS to process task switches, a task switch fault should also generate a task switch using the TSS.

## *25. Low Frequencies With 5:2 Core to Bus Clock Ratio May Fail in FRC*

**PROBLEM:** If a functional redundancy checking (FRC) system is using a 5:2 core to bus clock ratio, and is running at frequencies of 120/48 MHz or lower, the phase-locked loops (PLL) of the master-checker pair may not function correctly, resulting in the assertion of FRCERR with the first request to the Pentium Pro processor bus. If the first request does not result in the assertion of FRCERR, the system will function normally. The minimum core frequency specification in the *Pentium® Pro Processor Developer's Manual*, *Volume 1* is 100 MHz.

**IMPLICATION:** An FRC system should not be run at core frequencies lower than 120 MHz with a 5:2 core to bus clock ratio. If it is necessary to do so, the system may not always begin executing correctly, but power-cycling the system after such behavior may allow normal operation.

**WORKAROUND:** Run the system at its specified frequency, or power-cycle the system if it is necessary to run an FRC system under these conditions.

## *26. RDPMC Cannot Be Used in Conjunction With SMM*

**PROBLEM:** If the performance counter enable (PCE) bit (bit 8 of CR4) is enabled (set to '1'), and if an RSM instruction is executed, the processor will enter a shutdown state, believing that reserved space has been altered.

**IMPLICATION:** The PCE bit is set by an operating system to enable user code to issue an RDPMC (Read Performance Monitor Counter) instruction. In systems which support SMM, this bit should not be set, and this instruction will not be available. However, the performance monitoring capabilities of the Pentium Pro processor are still available through driver-based applications.

**WORKAROUND:** To prevent the shutdown from occurring, SMM handler code can clear this bit. However, this will disable the ability of users to issue the RDPMC instruction, and any application which attempts to issue this instruction after SMM handler code has been executed will cause a general protection fault, unless the operating system has re-enabled it by setting the PCE bit.

## *27. PWRGOOD Forced to 0 During Boundary Scan Resets TAP*

**PROBLEM:** According to IEEE 1149.1-1990 (JTAG) Specification, rule 5.3.1(b), no component input pin should initialize the test access port (TAP) controller, or any other test logic within the component. However, if the PWRGOOD pin on the Pentium Pro processor is forced to 0, it will reset the TAP Controller.

**IMPLICATION:** If boundary scan testing forces PWRGOOD to 0, boundary scan errors will result.

**WORKAROUND:** Do not force PWRGOOD to 0 during boundary scan testing. The BSDL file or the test vectors can be edited to ensure that this does not occur, by changing the declaration of the PWRGOOD pin from 'input' to 'internal' with a safe value of 1. Note that while this will allow boundary scan tests to be generated correctly, it will not allow the PWRGOOD pin to be tested.

## 28. BIST Failure Not Indicated When RUNBIST TAP Command Used

**PROBLEM:** The test access port (TAP) RUNBIST command is intended for use during boundary scan testing, and should run the built-in self test (BIST), storing the result in the BIST result register after completion. A zero result indicates BIST passed, and a nonzero result indicates BIST failure. Currently, if this command is executed from the TAP, a zero result will be stored in the BIST result register regardless of whether the BIST passes or fails.

**IMPLICATION:** Running the RUNBIST command from the TAP will not indicate a failing condition. This command is not functional for the steppings of the Pentium Pro processor affected by this erratum.

**WORKAROUND:** None identified. However, BIST may still be run separately from boundary scan, by allowing the INIT# signal to be sampled active on the RESET# signal's active to inactive transition. In this case, passage or failure is indicated in EAX.

## 29. INVLPG May Not Invalidate Targeted ITLB Entry

**PROBLEM:** When the INVLPG instruction is used, the entry in the translation look-aside buffer (TLB) which corresponds to the given address should be invalidated, under most circumstances preserving the validity of the rest of the TLB. However, if the variable memory type range registers (MTRRs) are used to map a region of memory as a particular memory type, this may not occur. Specifically, if the variable MTRR specifies a range which is not 4-Mbyte aligned (i.e. the MTRR's mask register contains a value with one or more bits in the range [21:12] nonzero), the INVLPG instruction will invalidate the proper entry in the data TLB (DTLB), but in so doing, may lose the information necessary to invalidate the entry in the instruction TLB (ITLB). It will, instead, invalidate entry 0 in the ITLB, leaving a stale translation in the ITLB. A subsequent code fetch from the addresses in this entry will retrieve and execute incorrect instructions.

**IMPLICATION:** The INVLPG entry cannot be used on a system with an MTRR range which is not 4-Mbyte aligned.

**WORKAROUND:** BIOS and operating system code should not use a variable MTRR range which is not 4-Mbyte aligned. All such ranges should have bits [21:12] of the MTRR's mask register set to '0' for processor steppings affected by this erratum.

## 30. SMI Does Not Flush TLB Entries With PGE Enabled

**PROBLEM:** When SMI# is asserted, all entries of the translation lookaside buffer (TLB) should be flushed. However, entries with the page global enable (PGE) bit set will not be flushed from the TLB.

**IMPLICATION:** The SMM (System Management Mode) handler may access an address which will be incorrectly translated due to the stale entries in the TLB, resulting in unexpected system behavior.

**WORKAROUND:** If a write to CR3 is executed to clear the TLBs upon entry into SMM, BIOS code can contain a workaround for this erratum.

## 31. I/O Restart in SMM May Fail After Simultaneous MCE

**PROBLEM:** If an I/O instruction (IN, INS, REP INS, OUT, OUTS, or REP OUTS) is being executed, and if the data for this instruction becomes corrupted, the Pentium Pro processor will signal a machine check exception (MCE). If the instruction is directed at a device which is powered down, the processor may also receive an assertion of SMI#. Since MCEs have higher priority, the processor will call the MCE handler, and the SMI# assertion will remain pending. However, upon attempting to execute the first instruction of the MCE handler, the SMI# will be recognized and the processor will attempt to execute the SMM handler. If the SMM handler

is completed successfully, it will attempt to restart the I/O instruction, but will not have the correct machine state, due to the call to the MCE handler.

**IMPLICATION:** A simultaneous MCE and SMI# assertion may occur for one of the I/O instructions above. The SMM handler will attempt to restart such an I/O instruction, but will have corrupted state due to the MCE handler call, leading to failure of the restart and shutdown of the processor.

**WORKAROUND:** If a system implementation must support both SMM and MCEs, the first thing the SMM handler code (when an I/O restart is to be performed) should do is check for a pending MCE (after writing to CR3 as per the workaround for Erratum #30). If there is an MCE pending, the SMM handler should immediately exit via an RSM instruction and allow the MCE handler to execute and restart the I/O instruction. If there is not, the SMM handler may proceed with its normal operation.

## *32. SMBASE Reset on INIT# or INIT_IPI*

**PROBLEM:** In systems which support System Management Mode (SMM), the default value for the SMBASE is 30000h. This value may be changed, particularly in MP systems, where processors' SMM spaces do not overlap. However, when an assertion of the INIT# pin or an initialization interprocessor interrupt (INIT_IPI) is received by a Pentium Pro processor, the value of SMBASE is reset to 30000h.

**IMPLICATION:** An assertion of INIT# or an INIT_IPI will clear any changes made to the SMBASE value.

**WORKAROUND:** It is possible that BIOS code may contain a workaround for this erratum, in systems which contain C-0 or subsequent steppings of Pentium Pro processor silicon. No workaround is available for the B-0 stepping of the Pentium Pro processor.

## *33. MCE Due to L2 ECC Error Gives L1 MCACOD.LL*

**PROBLEM:** If an ECC error occurs on an access to the Pentium Pro processor's L2 cache, the resulting machine check exception (MCE) will be logged with '01' in the LL field of MCACOD. This value indicates an L1 ECC error; the value should be '10'.

**IMPLICATION:** An MCE due to an L2 cache access will be improperly logged as due to an L1 error.

**WORKAROUND:** None identified.

## *34. INVLPG Does Not Invalidate Entire 0 to 4-Mbyte Region*

**PROBLEM:** When using large paging modes (mode C or mode B paging), a TLB entry may be created for the large page beginning at address 0. This page may cover the 0 to 2 or 4-Mbyte region. If the fixed memory type range registers (MTRRs) are used in conjunction with these paging modes, TLB entries for 4-Kbyte pages may also be created to map the lower 1-Mbyte region of memory. If the INVLPG instruction is used to invalidate one of these 4-Kbyte pages, the large page which covers the same region will not be invalidated; conversely, if the large page is invalidated using INVLPG (with an address from 1 to 2 or 4-Mbytes), the 4-Kbyte entries will not be invalidated.

**IMPLICATION:** Intel has not currently identified any software which is affected by this erratum. However, using INVLPG to invalidate regions from 0 to 1-Mbyte or from 1 to 4-Mbytes may result in the use of stale data if the other part of the 0 to 4-Mbyte range is expected to be invalidated when using 4-Mbyte pages (or 0 to 2-Mbyte if the software uses 2-Mbyte pages). Future operating systems may be affected by this erratum, but only if the 0 to 4-Mbyte page is invalidated using INVLPG. Since this page would most likely contain operating system kernel code, it is unlikely that such an operating system would be designed to invalidate this page.

**WORKAROUND:** Operating systems should not set up a 4-Mbyte page at address 0, unless writes to the CR3 register are performed to clear the TLB entries for this region (if the PGE bit in CR4 is set, this bit must be cleared prior to writing to CR3, to clear global pages). This will invalidate the entire TLB. If a 4-Mbyte page is used, and INVLPG instructions are used to invalidate this region, INVLPG instructions must be used repeatedly to invalidate the large page as well as any of the 4-Kbyte ranges which should be invalidated.

## *35. BINIT# Assertion During Snoop Hit May Cause Double Machine Check Exception*

**PROBLEM:** If the L1 cache has locked a modified cache line in response to an external snoop, then the assertion of BINIT# between the time ADS# is asserted for the snoop and the time that the implicit writeback data is transferred may cause a double machine check exception (MCE). The double MCE is encountered due to artificial internal parity errors generated by the BINIT# handler during its cleanup routines.

**IMPLICATION:** A BINIT# may cause the processor to shut down in response to the double MCE generated if the above conditions occur. IERR# will be asserted, and no further operations are possible until a hard reset is performed.

**WORKAROUND:** BINIT# is asserted upon detection of a catastrophic system condition. Leaving MCEs disabled will result in the BINIT# causing the CPU to enter shutdown as well. To avoid CPU shutdown, BINIT# observation can be disabled. However, a condition which would result in the generation of BINIT# may cause corrupt data to be propagated, and unexpected system behavior may occur. No workaround which would enable BINIT# to be recognized in this window without causing the CPU to shutdown has been identified.

## *36. Machine Check Exception Handler May Not Always Execute Successfully*

**PROBLEM:** An asynchronous machine check exception (MCE), such as a BINIT# event, which occurs during an access that splits a 4-Kbyte page boundary may leave some internal registers in an indeterminate state. Thus, MCE handler code may not always run successfully if an asynchronous MCE has occurred previously.

**IMPLICATION:** An MCE may not always result in the successful execution of the MCE handler, hanging the CPU.

**WORKAROUND:** If MCEs are not enabled, execution of the MCE handler will not occur. However, asynchronous MCEs usually occur upon detection of a catastrophic system condition. Leaving MCEs disabled will result in the condition which caused the asynchronous MCE instead causing the CPU to enter shutdown. Therefore, leaving MCEs disabled may not improve overall system behavior. No workaround which would guarantee successful MCE handler execution under this condition has been identified.

## *37. Double ECC Error on Read May Result in BINIT#*

**PROBLEM:** For this erratum to occur, the following conditions must be met:

- Machine check exceptions (MCEs) must be enabled.
- A dataless transaction (such as a write invalidate) must be occurring simultaneously with a transaction which returns data (a normal read).
- The read data must contain a double-bit uncorrectable ECC error.

If these conditions are met, the Pentium Pro processor will not be able to determine which transaction was erroneous, and instead of generating an MCE, it will generate a BINIT#.

**IMPLICATION:** The bus will be reinitialized in this case. However, since a double-bit uncorrectable ECC error occurred on the read, the MCE handler (which is normally reached on a double-bit uncorrectable ECC error for a read) would most likely cause the same BINIT# event.

**WORKAROUND:** Though the ability to drive BINIT# can be disabled in the Pentium Pro processor, which would prevent the effects of this erratum, overall system behavior would not improve, since the error which would normally cause a BINIT# would instead cause the machine to shut down. No other workaround has been identified.

## *38. RSM Cannot Return to HALT or SHUTDOWN in 32-Bit OS*

**PROBLEM:** If a processor which was previously operating in 32-bit protected mode has shut down or executed a HLT instruction, and is then targeted by a System Management Interrupt (SMI), the SMM handler will be executed in a different mode. Upon returning to the HALT or SHUTDOWN state via an RSM, the mode state will not be correctly updated back to 32-bit protected mode, potentially crashing the operating system.

**IMPLICATION:** Systems which support SMM may hang if software which executes HLT instructions is being run, or if the processor is expected to be able to recover from a SHUTDOWN state via software recovery which uses SMM.

**WORKAROUND:** For the HALT case, SMM handler code may clear the Halt Auto Restart bit in the SMM dump space (offset 7F02h), then subtract 1 from the saved EIP (offset 7FF0h). This will cause the HLT instruction to be re-executed upon completion of the RSM, which will allow 32-bit protected mode to be re-entered and the processor to successfully enter the HALT state. No workaround has currently been identified for the SHUTDOWN case; however, it is not guaranteed that an SMI will be properly serviced when in this state.

## *39. Accesses of Modified Data in DCU May Hang System*

**PROBLEM:** Under heavy bus traffic in a system with multiple snooping agents (including processors or DMA controllers), the following sequence of transactions may cause the system to hang:

1. A read-for-ownership cycle (a.k.a read and invalidate line) is issued to an address which is in Modified state in both the L2 cache and the data cache unit (DCU).
2. The DCU writes the data at that address back to the L2, so that it is no longer in Modified state in the DCU, only in the L2. It is now in Exclusive state in the DCU.
3. A read occurs to the same address, which is found in the DCU.
4. The line in the DCU is changed to the Modified state via a write to that address.
5. An external snoop from another processor or DMA controller occurs to the address, receives a HITM# (hit Modified) response, and the DCU's data goes out onto the processor bus.

At this stage, the internal state of the processor may be incorrect, causing a future transaction to hang the processor.

**IMPLICATION:** Heavy traffic in a system with devices capable of causing external snoops may cause the system to hang. However, Intel has observed the conditions necessary to cause the traffic described above only in a focus-testing environment.

**WORKAROUND:** None identified.

## 40. Branch Traps Do Not Function if BTMs Are Also Enabled

**PROBLEM:** If branch traps or branch trace messages (BTMs) are enabled alone, both function as expected. However, if both are enabled, only the BTMs will function, and the branch traps will be ignored.

**IMPLICATION:** These debugging features cannot be used together.

**WORKAROUND:** If branch trap functionality is desired, BTMs must be disabled.

## 41. Cache Line May Exist in Two Different Ways in MP System

**PROBLEM:** In heavy traffic on an MP system, it is possible for a cache line to be read into two different "ways" of the L2 cache, or for an invalid line to become spuriously validated. The conditions necessary for this behavior to occur are as follows:

- A read-for-ownership (a.k.a. read and invalidate line), normal read, and second read-for-ownership must occur, in that order, with no other intervening transactions.
- These three transactions must be pipelined together.
- They must be accessing the same set of the L2 cache.
- A cache line which is in the process of being changed to Shared state must be snooped by an external agent (such as another processor or DMA controller).

**IMPLICATION:** If these conditions occur, unexpected behavior will result. The most likely failure mechanism for this erratum is a processor hang, but it is possible that data corruption may result. Intel has observed this erratum only in a focus-testing environment with 3 or more processors in the system. Intel has not currently identified any software which causes this behavior.

**WORKAROUND:** Intel has been unable to reproduce this erratum with an IOQ depth of 1, as configured by the BIOS. This is a recommended configuration for the affected steppings of the processor; production systems with this configuration are not known to be affected by this erratum.

## 42. HALT, SHUTDOWN, and STPCLK Special Cycles Not Issued

**PROBLEM:** The HALT, SHUTDOWN, and STPCLK special cycles will not be issued by the Pentium Pro processor.

**IMPLICATION:** Systems which rely on these special cycles will not function correctly. No systems or software known to Intel rely on the processor issuing these cycles.

**WORKAROUND:** No known workaround exists for systems which use the B-0 stepping of the Pentium Pro processor or the A-2 stepping of the 82450GX/KX PCIset. Other systems may contain a workaround for this erratum in BIOS code.

## 1AP. APIC Access to Cacheable Memory Causes Shutdown

**PROBLEM:** APIC operations which access memory with any type other than uncacheable (UC) is illegal, and if machine check exceptions (MCEs) are disabled, the CPU will enter shutdown after such an access. If MCEs are enabled, an MCE will occur. However, in this circumstance, a second MCE will be signaled. This will also cause the Pentium Pro processor to enter shutdown.

**IMPLICATION:** Recovery from a PIC access to cacheable memory will not be successful. Correctly written software will not encounter this erratum.

**WORKAROUND:** Ensure that the memory space to which PIC accesses can be made is marked as type UC (uncacheable) in the memory type range registers (MTRRs) to avoid this erratum.

## *2AP. MP Systems May Hang Due to Catastrophic Errors During BSP Determination*

**PROBLEM:** In MP systems, a catastrophic error during the bootstrap processor (BSP) determination process should cause the assertion of IERR#. If the catastrophic error is due to the APIC data bus being stuck at electrical zero, then the system hangs without asserting IERR#.

**IMPLICATION:** MP systems may hang during boot due to a catastrophic error. This erratum has not been observed in a real system, but was found during focused system testing using a grounded APIC data bus.

**WORKAROUND:** None identified.

## *3AP. INIT_IPI After STARTUP_IPI-STARTUP_IPI Sequence May Cause AP to Execute at 0h*

**PROBLEM:** The MP Specification states that to wake up an application processor (AP), the interprocessor interrupt sequence INIT_IPI, STARTUP_IPI, STARTUP_IPI should be sent to that processor. On the Pentium Pro processor, an INIT_IPI, STARTUP_IPI sequence will also work. However, if the INIT_IPI, STARTUP_IPI, STARTUP_IPI sequence is sent to an AP, an internal race condition may occur in the APIC logic which leaves the processor in an incorrect state. Operation will be correct in this state, but if another INIT_IPI is sent to the processor, the processor will not stop execution as expected, and will instead begin execution at linear address 0h. In order for the race condition to cause this incorrect state, the system's core to bus clock ratio must be 5:2 or greater.

**IMPLICATION:** If a system is using a core to bus clock ratio of 5:2 or greater, and the sequence INIT_IPI, STARTUP_IPI, STARTUP_IPI is generated on the APIC bus to wake up an AP, and then at some later time another INIT_IPI is sent to the processor, that processor may attempt to execute at linear address 0h, and will execute random opcodes. Some operating systems do generate this sequence when attempting to shut the system down, and in a multiprocessor system, may hang after taking the processors offline. The effect seen will be that the OS may not restart the system if 'shutdown and restart' or the equivalent is selected upon exiting the operating system. If an operating system gives the user the capability to take an AP offline using an INIT_IPI (Intel has not identified any operating systems which currently have this capability), this option should not be used.

**WORKAROUND:** BIOS code should execute a single STARTUP_IPI to wake up an application processor. Operating systems, however, will issue an INIT_IPI, STARTUP_IPI, STARTUP_IPI sequence, as recommended in the MP specification. It is possible that BIOS code may contain a workaround for this erratum in systems with C-0 or subsequent steppings of Pentium Pro processor silicon. No workaround is available for the B-0 stepping of the Pentium Pro processor.